DETAILED ACTION

1. This is a non-final Office Action in response to the present US application number 10/540,186 filed on June 20th 2005 and IDS filed on June 20th 2005. Claims 1-27 are pending and have been examined.

Specification

2. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification. For example:

- Instances of (19,20,22,26,30) should be replaced with (19, 20, 22, 26, and 30).
- Instances of (14,34,36) should be replaced with (14, 34, and 36).

Appropriate correction is required

Claim Objections

3. **Claims** are objected to because of the following informalities:

- C**laim 13** is objected to because there is no antecedent basis for the populating step. It is recommended that claim 13 be amended to be dependent on claim 12. Further actions are based on this assumption.

- **Claim 25** is objected to because it is dependent upon a claim that succeeds the claim. It is recommended that claim 25 be amended to be dependent on claim 23. Further actions are based on this assumption.

Appropriate corrections are required.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. **Claims 1, 2, 6, 10-12, 14-18, 22, 26, and 27** are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent Publication No. 2004/0078487 A1 to *Cernohous et al. ("Cernohous")*.

As to **claim 1**, *Cernohous* discloses a method of establishing communication between a client node (14, 34, and 36) and a server node in a heterogeneous IP network (300), said method comprising the steps of:

making a request, by a user associated with said client, to a portal (18) in said network (300) for a list of server hostnames (51) capable of providing a desired content to said client. In particular, *Cernohous* further discloses a

method wherein a query is formulated on one of the clients. Then if it's determined that the response to the query is not in the cache, the resolver queries the external DNS, which would have a list of server hostnames. The external DNS can be interpreted as the portal (paragraph [0043]; Figs. 2, 4, 5, 8, and 9);

providing a first table (50) and a second table (40) from said portal (18) to said client responsive to said client request, said first table (50) including said list of server node hostnames (51). In particular, *Cernohous* further discloses of a table within the client's internal cache and a table within the DNS (Figs. 2, 4, 5, and 8; 822);

filtering at said client (14, 34, and 36), said provided list of server hostnames (51) to exclude those server hostnames (51) with whom said client (14, 34, and 36) cannot establish a communication. In particular, *Cernohous* further discloses of the concept where each cached response has a related time to live (TTL). The response is valid as long as the TTL value is still positive. Otherwise, the cache is filtered and the new response is queried from the DNS along with a new TTL to that response (paragraphs [0030] and [0031]);

selecting by said user, a server hostname (51) from said filtered list of server hostnames (51). In particular, *Cernohous* further discloses of a table that gets refreshed or filtered after each query (Figs. 4 and 5) using a round robin approach to deliver the IP addresses to achieve dynamic load balancing (paragraph [0008], Figs. 4 and 5);

determining from said first table (50) if an IP address associated with said user selected server hostname (51) is resolvable via a domain name server (DNS). In particular, *Cernohous* further discloses when a client needs to determine an IP address that corresponds to a domain name, the client formulates a query and submits the query to resolver (paragraph [0030]; Fig. 7);

if said step (e) is satisfied, obtaining said associated IP address from said DNS. In particular, *Cernohous* further discloses the resolver would respond by checking its cache to see if a valid response is still alive, depending on the TTL, and if so, the response is returned to the requesting client (paragraph [0030]). In addition, when there's no valid response in the cache of the client. The resolver then sends a query to the DNS for a match (paragraph [0031]); and

if said step (e) is not satisfied, executing a protocol by said client (14, 34, and 36) with said portal (18) to determine one or more default IP addresses of a server having said selected server's hostname (51). In particular, *Cernohous* further discloses of a procedure where the local DNS may include a cache that contains answers obtained from other DNSs. If the local DNS code does not know the answer to a query, it may generate a query to a different DNS, which may generate a query to yet another DNS until a DNS with the answer is located (paragraphs [0031] and [0032]).

As to **claim 2**, the rejection of claim 1 is incorporated and *Cernohous* further discloses the step of establishing a communication with said selected

server (19, 20, 22, 26, and 30) using said associated IP address, following said step (f). In particular, *Cernohous* further discloses when an application queries a DNS for [a domain name like] ibm.com, the DNS returns the IP addresses that correspond to the ibm.com web site (paragraph [0006]). In addition, *Cernohous* also discloses that usually there is only one IP address returned from a query and that is what gets used by the software application (paragraph [0008]).

As to **claim 6**, the rejection of claim 1 is incorporated and *Cernohous* further discloses the step of establishing a communication with said selected server (19, 20, 22, 26, and 30) using said associated IP address, following said step (g). In particular, *Cernohous* further discloses of a procedure where the local DNS may include a cache that contains answers obtained from other DNSs. If the local DNS code does not know the answer to a query, it may generate a query to a different DNS, which may generate a query to yet another DNS until a DNS with the answer is located (paragraphs [0031] and [0032]).

As to **claim 10**, the rejection of claim 1 is incorporated and *Cernohous* further discloses wherein said determining step further comprises performing a lookup in said first table (50) by said client (14, 34, 36) using said user selected server hostname (51) as an index, to obtain a record value indicating the resolvability status of the selected server's hostname (51) via said DNS. In particular, *Cernohous* further discloses of the concept of the configuration files

(264) that define the relationship between domain names and their corresponding network addresses. Each resource record in configuration files may specify a corresponding TTL (paragraph [0027]). So, from time to time, the values within the DNS may become invalid too when the TTL expires. This could be interpreted as a resolvability status indicator. In addition, the DNS may include a cache that contains answers obtained from other DNSs. It may generate a query to different DNS, which may generate a query to yet another DNS until a DNS with the answer is located (paragraph [0032]).

As to **claim 11**, it is the same method claims corresponding to method claim 6 with the only change being instead of using the first table, now it is using a second table. As such, all of claim 11 is also rejected under the same reasons set forth in connection with the rejections of claim 6.

As to **claim 12**, the rejection of claim 1 is incorporated and *Cernohous* further discloses prior to said step (a), the step of populating said first table (50) at said vendor portal (18) with a plurality of records, each of the records comprising:

said server hostname (51) of a server in said network (300) capable of providing said desired content to said client (14, 34, and 36). In particular, *Cernohous* further discloses in an exemplary embodiment where the queried DNS can provide the hostname, which in this case is ibm.com (Figs. 4 and 5);

an IP address version (53) associated with said server hostname (51). In particular, *Cernohous* further discloses in an exemplary embodiment where the queried DNS can provide the IPv4 addresses associated with the hostname, ibm.com (Figs. 4 and 5);

and an indicator (55) of whether said server hostname (51) is resolvable via a DNS server in said network (300). In particular, *Cernohous* further discloses of a TTL marker corresponding to each response in the cache, which could be used as an indicator of whether the hostname entry is resolvable or not (paragraph [0027]). In addition, see rejection of claim 10.

As to **claim 14**, the rejection of claim 1 is incorporated and *Cernohous* further discloses prior to said step (a), the step of populating said second table (40) at said vendor portal (18) with a plurality of records, each of the records comprising:

said server hostname (41) of a server in said network (300) capable of providing said desired content to said client (14, 34, and 36). In particular, *Cernohous* further discloses in an exemplary embodiment where the queried DNS can provide the hostname, which in this case is ibm.com (Figs. 4 and 5);

a default IP address (43) associated with said server hostname (41). In particular, *Cernohous* further discloses in an exemplary embodiment where the queried DNS can provide the IPv4 addresses associated with the hostname, ibm.com (Figs. 4 and 5); and

a relay router address (45). In particular, *Cernohous* further discloses of a multiple address cache mechanism. In an exemplary embodiment as depicted in Figures 4 and 5, each hostname can have multiple IP addresses associated with it (Figs. 4 and 5). As such, an alternate IP address may be interpreted as the relay router address.

As to **claim 15**, the rejection of claims 1 and 14 are incorporated and *Cernohous* further discloses wherein said populating step is performed during a registration stage prior to the operation of said IP network (300). In particular, *Cernohous* further discloses of an internal cache and a resolver within the client. Due to the short TTL value associated with each response, the DNS is queried by the resolver (Figs. 1 and 8; paragraphs [0030] and [0031]). This refreshes all the response entries within the cache of the client with new TTL values and can be interpreted as a populating step.

As to **claim 16**, the rejection of claim 1 is incorporated and *Cernohous* further discloses wherein said step (c) further comprises the steps of:

comparing an IP address version of said client (14,34,36) with one or more IP address versions (53) associated with said server hostname (51) to determine from said comparison if said compared IP address versions are capable of establishing a communication between said client (14,34,36) and said server; In particular, *Cernohous* further discloses of the resolver within the cache

of the client system which can query the DNS to compare the versions of the IP address (paragraph [0030]).

if said comparison step is satisfied, retaining said server hostname (51) and associated record information in said filtered list. In particular, *Cernohous* further discloses that using the TTL value, the value of the cached response is valid as long as the TTL value is still positive (paragraph [0030]).

and otherwise deleting said server hostname (51) and said associated information from said filtered list. In particular, *Cernohous* further discloses when the TTL value has expired; the DNS is queried by the resolver for a new updated value to be entered into the cache (paragraphs [0030] and [0031]).

As to **claims 17 and 18,** they are the same system claims corresponding to method claims 1 and 2 respectively. As such, all of claims 1 and 2 are also rejected under the same reasons set forth in connection with the rejections of claims 1 and 2 respectively.

As to **claim 22**, it is the same system claim as claim 6. As such, all of claim 22 is also rejected under the same reasons set forth in connection with the rejection of claim 6.

As to **claims 26 and 27**, they are the same system claims corresponding to method claims 12 and 14. As such, all of claims 26 and 27 are also rejected

under the same reasons set forth in connection with the rejections of claim 26 and 27.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. **Claims 3-5, 7-9, 13, 19-21, and 23-25** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Publication No. 2004/0078487 A1 to *Cernohous* in view of U.S. Patent No. 7,321,598 B2 to *Blanchet et al.* ("*Blanchet*").

As to **claim 3**, the rejections of claims 1 and 2 are incorporated and *Cernohous* further discloses wherein the step of establishing a communication with said selected server (19, 20, 22, 26, and 30) further comprises the steps of:

determining if an IP version of a first returned IP address from said DNS is the same version as an IP version of said client (14, 34, and 36). In particular, *Cernohous'* concept of querying the DNS only deals with one version, mainly just IPv4 address (Figs. 4 and 5). As such, *Cernohous does* not expressly disclose about this determining step of checking if the versions of the client match the IP address obtained from the DNS.

Blanchet discloses more expressly about how a tunnel client sends the version of the [tunnel setup protocol] TSP that it supports using the control

channel to the tunnel broker server 60 (102). On receipt of the TSP protocol version, the tunnel broker server determines whether it supports the same version of the tunnel setup protocol (column 5, lines 49-54). In addition, *Blanchet* discloses later on that each tunnel and its tunnel setup protocol is associated with its own unique IP address (column 10, lines 33-36).

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of checking the compatibility versions between a client and a server in case of multiple versions with *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36);

if said step (1) is not satisfied, obtaining a next returned IP address from said DNS and repeating said step (1). In particular, *Cernohous* does not expressly disclose about this step.

Blanchet discloses more expressly wherein the tunnel broker server determines whether it has an alternative list of tunnel broker servers that it can provide to the tunnel client (column 5, lines 58-61). In essence, this is iterating

step 1 until a suitable list is found and to establish a compatible connection between the client and the server.

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of repeatedly checking the compatibility versions between a client and a server until a match along with *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36);

if said step (1) is satisfied, determining if said IP version of said first returned IP address and said IP version of said client (14, 34, and 36) are IPv6 versions. In particular, *Cernohous* does not expressly disclose about this step.

Blanchet discloses more expressly wherein the client node in the beginning was IPv6 as depicted in its illustrations (Figs. 4 and 6; 72). Ultimately, the whole concept was to get an IPv6 device across an IPv4 network (column 3, lines 34-36).

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of checking the compatibility versions between a client and a server into *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36);

if said step (3) is satisfied, determining if said IPv6 versions are 6 to 4 addresses. In particular, *Cernohous* does not expressly disclose about this step.

Blanchet discloses more expressly wherein the tunnel client configures a tunnel endpoint referred to as the tunnel client endpoint for the IPv6-in-IPv4 tunnel (column 3, lines 49-52). Ultimately, Blanchet's concept is to be able to transport IPv6 addresses across an IPv4 network, using IPv6-in-IPv4 tunneling.

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of checking the IPv6 addresses to make sure they are 6 to 4 addresses with *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can

handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36); and

if said step (4) is satisfied, establishing a communication with said selected server (19, 20, 22, 26, and 30) using said IPv6 protocol and automatic tunneling. In particular, *Cernohous* does not expressly disclose about this

Blanchet discloses more expressly wherein after setting up the IPv6-in-IPv4 tunnel; it would permit the automated establishment of IPv6-in-IPv4 tunnels using a control channel (column 3, lines 49-52).

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of using IPv6-in-IPv4 tunneling with *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36);

As to **claim 4**, the rejections of claims 1, 2, and 3 are incorporated and *Cernohous* further discloses wherein if said step (4) is not satisfied, establishing a communication with said selected server (19, 20, 22, 26, and 30) using an IPv6

protocol and a tunneling method having a relay router as an endpoint address. In particular, *Cernohous* does not expressly disclose this.

Blanchet discloses more expressly wherein the tunnel broker server endpoint may be supported by the tunnel broker server, or by another gateway node, such as an IPv4/IPv6 router connected to both the IPv4 and IPv6 networks (column 3, lines 45-48). In addition, the illustration in Figure 6 also depicts the router towards the end of the tunnel (Fig. 6; 76).

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of using an IPv6 tunneling protocol method and having the router at the endpoint with *Cernohous'* overall concept of manipulating the TTL value within the internal cache and the DNS. The suggestion/motivation to combine them would be such that the DNS can handle both IPv4 and IPv6 queries and enable a IPv6 device to communicate across an IPv4 network (*Blanchet*: column 3, lines 34-36);

As to **claim 5**, the rejections of claims 1, 2, and 3 are incorporated and *Cernohous* further discloses wherein if said step (3) is not satisfied, establishing a communication with said selected server (19, 20, 22, 26, and 30) using an IPv4 protocol. In particular, *Cernohous* further discloses of the steps after determining

the compatibility of the IP addresses wherein an exemplary embodiment, when the query is sent out to the DNS, the IP addresses within the domain of ibm.com are all following IPv4 protocol (paragraph [0031]; Figs. 4 and 5). However, *Cernohous* does not expressly disclose of this step of determining whether the IP versions of the address are the same as the client beforehand.

Blanchet discloses more expressly about how a tunnel client sends the version of the [tunnel setup protocol] TSP that it supports using the control channel to the tunnel broker server 60 (102). On receipt of the TSP protocol version, the tunnel broker server determines whether it supports the same version of the tunnel setup protocol (column 5, lines 49-54). In addition, *Blanchet* discloses later on that each tunnel and its tunnel setup protocol is associated with its own unique IP address (column 10, lines 33-36).

Cernohous and *Blanchet* are analogous art because they are from the same field of endeavor with respect to solving compatibility issues between IPv4 and IPv6 protocols.

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to combine *Blanchet's* concept of checking the compatibility of the IP versions between those in a client and those in a server along with *Cernohous'* concept of using the common IPv4 standard to establish communication. The suggestion/motivation to combine them would be such that checking for the compatibility before establishing connections and

communications would lead to less problems afterwards (*Blanchet*: column 5, lines 49-54; column 10, lines 33-36);

As to **claims 7, 8, and 9**, they are the same method claims corresponding to method claims 3, 4, and 5 respectively with the only change being instead of using the first table, now it is using a second table. As such, all of claims 7, 8, and 9 are also rejected under the same reasons set forth in connection with the rejections of claims 3, 4 and 5 respectively.

As to **claim 13**, *Cernohous* further discloses wherein said populating step is performed during a registration stage prior to the operation of said IP network (300). In particular, *Cernohous* further discloses of an internal cache and a resolver within the client. Due to the short TTL value associated with each response, the DNS is queried by the resolver (Figs. 1 and 8; paragraphs [0030] and [0031]). This refreshes all the response entries within the cache of the client with new TTL values and can be interpreted as a populating step.

As to **claims 19-21, 23, and 24** they are the same system claims corresponding to method claims 3-5, 7, and 8 respectively. As such, all of claims 3-5, 7, and 8 are also rejected under the same reasons set forth in connection with the rejections of claims 3-5, 7, and 8 respectively.

As to **claim 25**, it is the same system claim as claim 9. As such, all of claim 25 is also rejected under the same reasons set forth in connection with the rejection of claim 9.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to XIANG YU whose telephone number is (571)270-5695. The examiner can normally be reached on Monday - Friday 8:00am - 5:00pm with every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Derrick Ferris can be reached on (571)272-3123. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/X. Y./
Examiner, Art Unit 4127

/Derrick W Ferris/
Supervisory Patent Examiner, Art Unit 4127